2. TERMINOLOGY FOR GCCS

This section provides:

A discussion of why common terminology is important

Definitions of GCCS-related architecture types, drawing types, and system components.

WHY COMMON TERMINOLOGY IS NEEDED

Why do definitions of architecture terms occupy a major section of this guidebook, instead of being included in a glossary at the back? Why do we need to worry about definitions at all?

The obvious reason is that the functionally and geographically diverse members of the GCCS community all need to "speak the same language." Terms relevant to the GCCS architecture need to be defined right up front because, in many cases, there is no unanimous agreement on what the terms mean. For example, architecture experts may have a rigid and narrow concept of what a "technical architecture" is and is not, while others may have a fuzzier idea of what the phrase means and may use it differently. We need to establish a clear definition of this and other terms to serve as a common frame of reference for all personnel involved in developing, evaluating, and implementing GCCS architectures.

Another reason for emphasizing terminology at this point is to promote the use of GCCS concepts, as well as GCCS vocabulary. Certain phrases appearing in this section, such as some of the specialty architecture types and drawing classifications, describe GCCS-specific information management practices. This section introduces these concepts, for which detailed application guidance appears in Section 4.

SOURCES OF COMMON TERMS

Department of Defense (DoD)-wide reference documents are the primary source of the terminology provided in this section. These references, such as the Technical Architecture Framework for Information Management (TAFIM), trace the architecture environment from the highest level through intermediate levels down to the technical documents that provide supporting information for the architect and engineer. These references are listed in Appendix E.

Where DoD guidance does not supply an authoritative definition for a phrase used in this section, industry sources and academic texts have been consulted. In many cases, the challenge has not been to find a definition for each phrase, but rather to reconcile a number of different definitions for the same term, or multiple terms conveying the same meaning. Applicability to GCCS architecture is the principal criterion driving the selection of specific terms and definitions presented here.

ARCHITECTURE TERMS: TYPES OF ARCHITECTURE DEPICTIONS

DoD policies require agencies responsible for defense information system resources to address requirements definition, design, procurement, implementation, integration, maintenance, sustainment, and disposition throughout the entire information system life cycle. Architecture planning and depiction is the foundation for action in each of these life-cycle phases.

As GCCS moves from initial operating capability to wide usage as the command and control system for United States (U.S.) forces, different architecture views are needed to support installation, migration activities, maintenance, and other tasks. These views present different “slices” of the overall GCCS architecture, at varying degrees of scope and specificity. The architecture depictions that GCCS users will receive, generate, or contribute to are described in the following section.

GCCS Architecture Types: Core , Objective, and Specialty

As the need to view data in different “slices” and from different perspectives increases, and as the automated capability to do so expands, the number of distinct architecture views continues to grow. This document distinguishes among *core architectures* – the essential set for which there are generally accepted definitions and boundaries in the architecture community, *objective architectures* – which define long-term policy and migration from the currently fielded systems, and *specialty architectures*, which show many specialized, less common views of the GCCS architecture. Figure 2-1 lists the depictions belonging to each of these categories.

GCCS Architecture Types

Core:

- Functional Architecture
- Technical Architecture
- System Design/Installation/ Implementation Architecture

Objective:

- Strategic Architecture
- Target Architecture
- Baseline Architecture

Specialty:

- Organizational Architecture
- Communications Architecture Hierarchical Information Flow
- Deployment Architecture
- Geographical Architecture
- Representational Architecture
- Security Architecture
- Standard Operating Procedure Architecture
- Reference Architecture
- Template Architecture
- On-line, Near-Realtime Architecture for Configuration Management

Figure 2-1.
GCCS Architecture Types

The definitions included in this section for the core and objective architectures capture the terminology used by the GCCS Architect, and are consistent with the TAFIM, the DoD 5000 series, and other DISA guidance, as well as accepted industry practice. The specialty architectures are less formalized. Some, such as the geographical and security architectures, capture subsets of information that architects, engineers, and other personnel frequently want to see in detail. Other specialty architectures represent relatively new ideas: architects may not be accustomed to generating these views and may not recognize them by name. Nevertheless, these specialty depictions highlight important subsets or "slices" of architecture data and can facilitate planning, configuration management, and problem-solving.

!

An important point: Although a large number of architecture types are defined in this section, each site does not necessarily require each and every one. Most sites will need to generate and maintain the core architectures for GCCS, and you may create or be given objective architectures to guide future actions. You should look over the specialty architecture depictions identified in this section and adopt only those that support individual task areas and add value to your work.

Core Architectures

The core architecture types reflect an evolution of data from the general to the specific, and from the missions and functions to be supported to the systems that support them. The functional, technical, and system design/installation/implementation architectures are typically viewed as interlocking, sequential pieces.

Functional Architecture. According to DoD 5000.1-M and the TAFIM, a functional architecture identifies the major functions or processes an organization performs and the operational interrelationships among those processes. The DISA Architecture Requirements and Definitions (ARD) report states that a functional architecture can be used to capture operational capabilities, information relationships among organizational elements, the assets supporting a particular function, the data created or used by each function, and the personnel who need and produce information.

A functional architecture gives an organization an understanding of its essential functions and the interrelationship of GCCS-supported activities and strictly local tasks. A functional architecture can help planners evaluate priorities, ensure support for critical capabilities, and improve the interoperation of GCCS and local systems.

Technical Architecture. A technical architecture defines the information technology environment that supports the work processes identified in the functional architecture. A technical architecture is a framework interrelating the data, computer platforms, applications, communications, and interfaces used by an organization and linking that organization to the external environment.

A technical architecture focuses attention on the services required by the organization. Services include such things as data interchange, access control, database management, and multimedia capabilities. A technical architecture is the place to specify the standards to be complied with in each service area. By focusing on the services required by the organization rather than jumping ahead to a specific technical solution, the user minimizes the risk of selecting a component that does not really satisfy organizational needs and may not fit with the larger system.

System Design/ Installation/ Implementation Architecture. This framework provides the detailed structure of information systems, identifying their specific physical components and the connectivity among them. Computer specifications, wiring types, and connectivity with peripherals such as printers and modems are representative of the appropriate level of detail. This architecture may be broken out into several iterative depictions to illustrate phased implementation.

The practical purpose of this architecture is to ensure that system components work together on the local level, and that the system as a whole interoperates with other systems throughout the defense hierarchy. If you are solely in charge of setting up routers and gateways, while another person focuses on hooking Disk Operating System (DOS) clients up to the network, adherence to a complete system design/installation/implementation architecture can help to ensure that these individual efforts mesh together successfully.

?

But what about the physical, or topological, or applications architecture? There are many types of architecture beyond those listed here, and various groups would lobby for one or another of them to be among the core set for GCCS. All of these architectures are valid, and if they are useful for you, by all means continue to use them. The core group presented here is simply intended to satisfy common GCCS architecture documentation requirements while remaining comprehensible to nonarchitects. If you can generate and understand the three types in the preceding section, you can manage GCCS architecture data.

Objective Architectures

Objective architectures support migration from a current configuration to an objective, or goal, information system. These depictions reflect long-term requirements based on high-level policy or managerial guidance.

The target architecture and baseline architecture are commonly paired, with the baseline representing the organization's current status relative to the target. The strategic architecture, which may accompany the target and baseline in some uses, is a high-level depiction of a functional goal that is translated into more technical planning in the target architecture.

Strategic Architecture. A strategic architecture captures policy-driven guidance and provides overall, long-term direction for systems development and integration. This depiction shows a high-level view of a notional system, identifying key functional concepts, objective capabilities, and relationships among organizations. In practice, a strategic architecture is a means for "philosophers" or policy-makers to sketch out the essential elements of a technological infrastructure that will support a strategic goal. An example is the notional C4I for the Warrior architecture, which provides the technological underpinnings for the JTF warfighting concept.

Target Architecture. A target architecture models an objective information system architecture that will support desirable functional goals. The target is usually placed three to five years in the future. In describing a future objective, a target architecture may include components or technologies that are not yet available, but are anticipated to become feasible by the target time period.

A target architecture's value is based on its use as a guide for current decisions and actions: when system improvements or replacements are contemplated, changes should be evaluated in terms of how the proposed functionality supports that specified in the target architecture. Design and implementation activities should be undertaken with the intent of transforming the current configuration, step by step, to the target. Proposed changes with little longevity (i.e., those that do not fit with the long-term plan) can be evaluated in terms of the immediate solution they offer versus the cost of replacing them with a more interoperable technology later. The target architecture thus provides reliable guidance for making programmatic and fiscal decisions.

Establishing a clear goal for the future also helps to ensure that separate efforts support a shared goal. With a target architecture in place, different individuals at different times (e.g., you and your successor, when you move on to greener pastures) may take actions with confidence that changes will be consistent with the long-range information system plan.

Baseline Architecture. A baseline architecture documents a site or organization's existing information system components and the connectivity among them. The practical purpose of a baseline architecture is to compare the current status shown in the baseline with the target architecture to highlight "deltas" between the two and deficiencies in current capabilities. The data for developing the baseline architecture is typically obtained from on-site examination of equipment and documentation.

Specialty Architectures

Specialty architectures provide a way to present information that is not captured in the core or objective architectures.

Specialty architectures do more than isolate or magnify a specific part of an architecture. (For a discussion of how to declutter a drawing to focus on certain details, see "Thematic Layering" in Section 4.) Specialty architectures provide unique information by adding a layer of technical or managerial annotation, or by interpreting or summarizing architectural data.

The following nine architecture types support anticipated programmatic and technical needs of GCCS developers and users. If you find it helpful to structure information in a way that is not described here, feel free to develop additional architectural views or adapt the ones described here. Conversely, there is no requirement whatsoever, either from the GCCS Architect or from a practical standpoint, to draw up each of the architecture views described in this guidebook. The essential point of preparing architectures is not to impress people with exciting graphics, but to gather relevant information and present it in a way that optimally supports planning, management, implementation, or another legitimate organizational need.

A graphic example, greatly simplified, is provided for each architecture type. These are merely “sketches” to illustrate the concepts discussed in the text; they are not meant to show the standard formats and conventions for GCCS architecture drawings provided in Section 3.

Organizational Architecture. A basic organizational chart – the familiar drawing with boxes and lines showing who reports to whom – can be considered an organizational architecture. In the defense architecture context, brief mission statements are often added to the basic chart to identify the functional role of each "box" (which may represent an individual or a group). This depiction supports the logical decomposition or breaking down of an organization's mission into smaller parts to be supported at various levels of the hierarchy. See Figure 2-2.

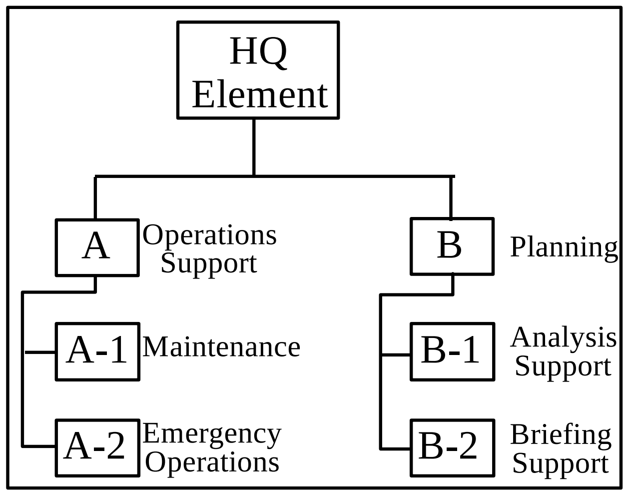


Figure 2-2.
Organizational Architecture – Concept Example

Communications Architecture/ Hierarchical Information Flow. This architecture documents the lines of communication among organizational elements and shows the flow of information between those elements. This view is distinct from the organizational architecture in that it shows the actual flow of information required to perform a function, not just a formal chain of command. The Integration Definition (IDEF) method is one way, though certainly not the only way, to model and structure the information for this architecture type.

In most cases it is not possible to create a single drawing to depict all of the information flows within an organization; generally this type of drawing will focus on a particular function (e.g., message handling). If desired, high-level elements of the telecommunications infrastructure can be included in the drawing to indicate how communication is achieved among geographically dispersed elements See Figure 2-3.

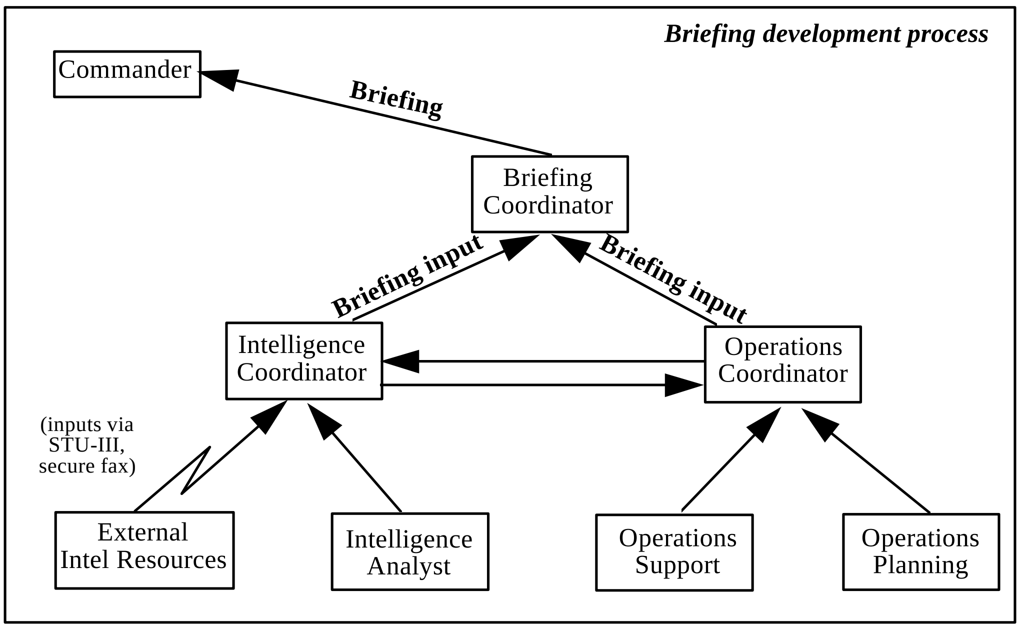


Figure 2-3.
Communications Architecture/Hierarchical Information Flow – Concept Example

Deployment Architecture. This architecture identifies assets and resources in terms of being fixed, fixed-to-deployable, or fully deployable. Depending on the purpose for which it is being developed, this architecture can show the connectivity between mobile and fixed assets; provide the location of deployed resources; highlight key interfaces between the strategic, theater tactical, and the forward deployed echelons; and/or the general communication connectivity (long haul communications, and theater tactical equipment) with the interfacing gateways. See Figure 2-4.

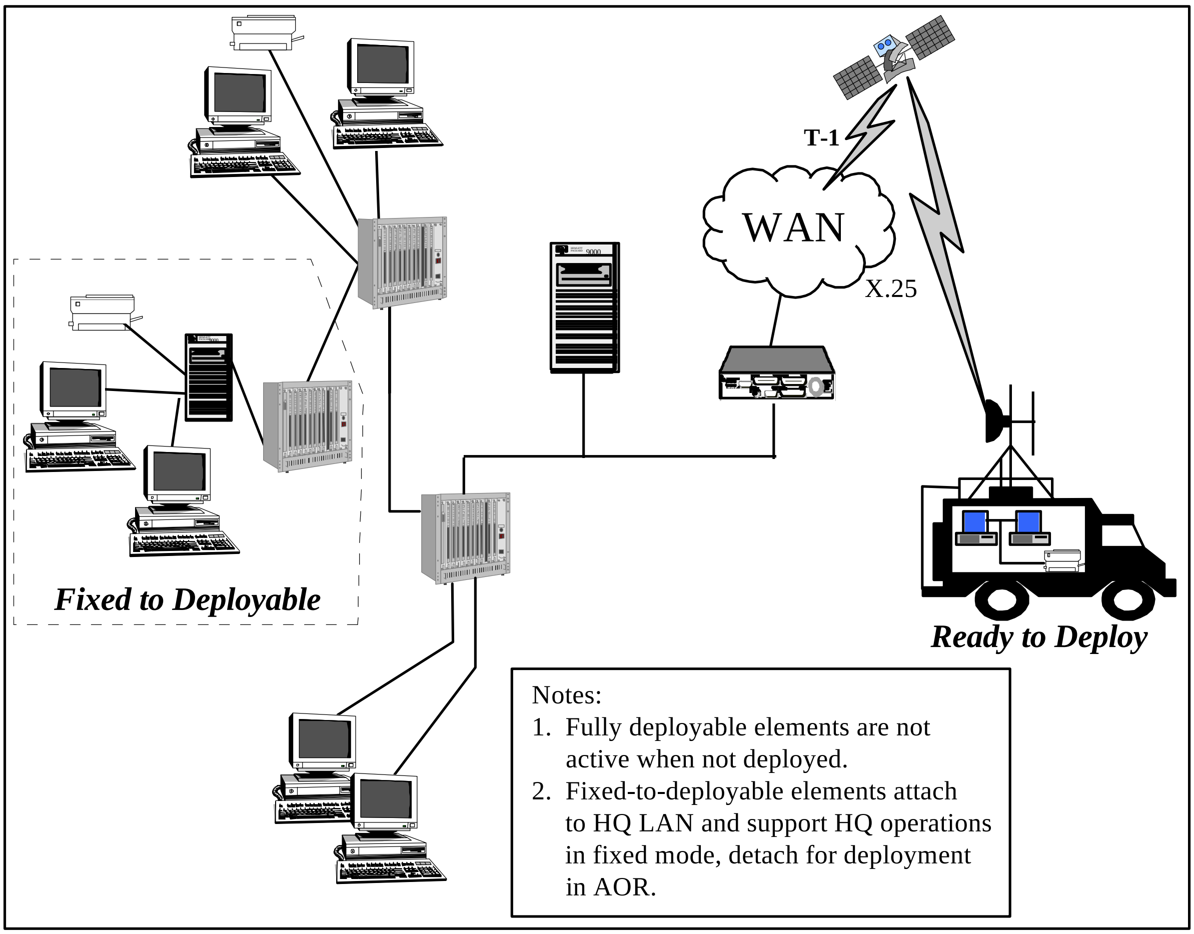


Figure 2-4.
Deployment Architecture – Concept Example

Geographical Architecture. This architecture depicts the information systems' physical locations and their connectivity from a geographic perspective. The focus here is on the locations of elements, the distances between them, and other key geographical information. The geographical architecture identifies the systems that connect one region with another, the supporting telecommunications, and any other important factors relating to location. See Figure 2-5.

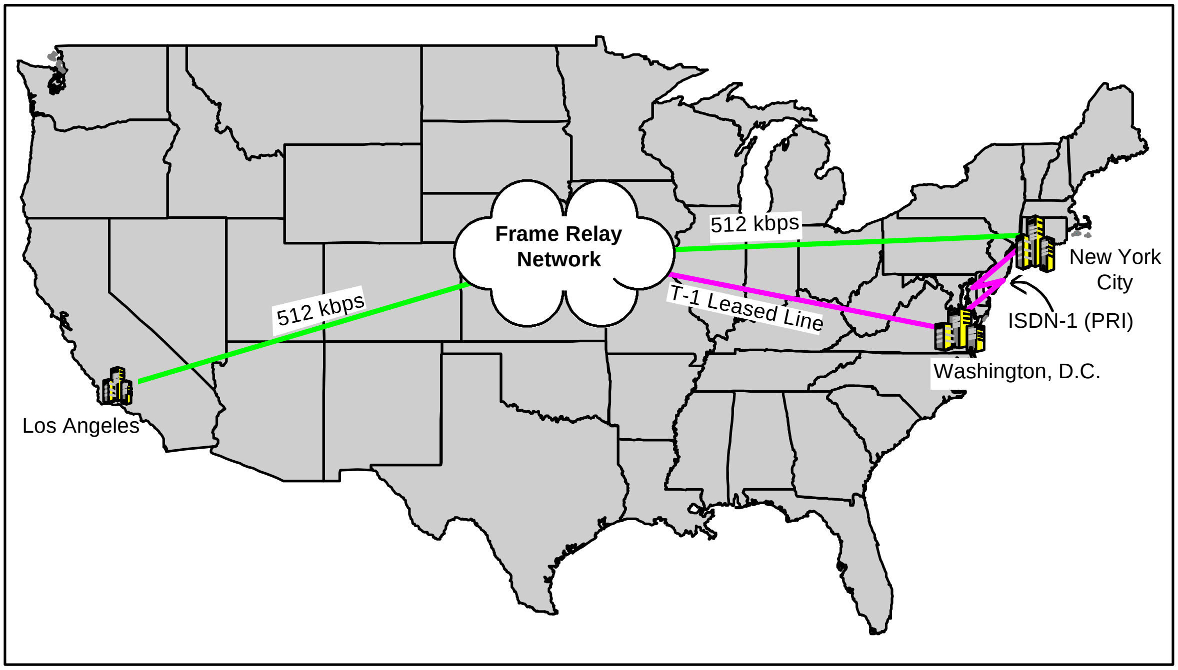


Figure 2-5. Geographical Architecture – Concept Example

Representational Architecture. A representational architecture provides a high-level conceptual depiction of a system. This type of architecture serves two distinct purposes.

First, a representational architecture can be used to simplify the depiction of complex systems by showing only the utilitarian parts of the architecture and restricting the level of detail. This may be done to make a complex system easier to understand, to focus attention on the "big picture," or to illustrate how technological elements support a policy, program, philosophy, or strategy.

The second use of a representational architecture is to communicate a general concept of operation or system design as the starting point for system development. Many concepts originate at the representational level (i.e., on the back of an envelope), and are later developed into technical architectures to support detailed engineering and research plans. See Figure 2-6.

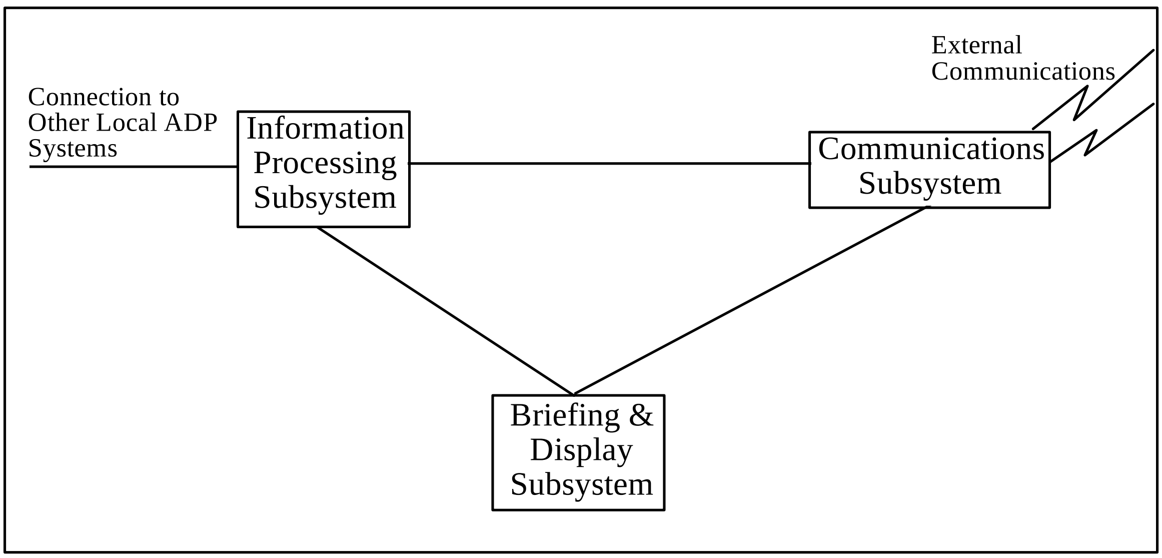


Figure 2-6.
Representational Architecture – Concept Example

Security Architecture. This architecture focuses on the capability of systems and networks to store, process, and transmit classified data. It highlights the classification status of components and the measures taken to safeguard information, such as firewalls and air-gaps. A security architecture can also be used to plan future security configurations, such as the implementation of multilevel security. See Figure 2-7.

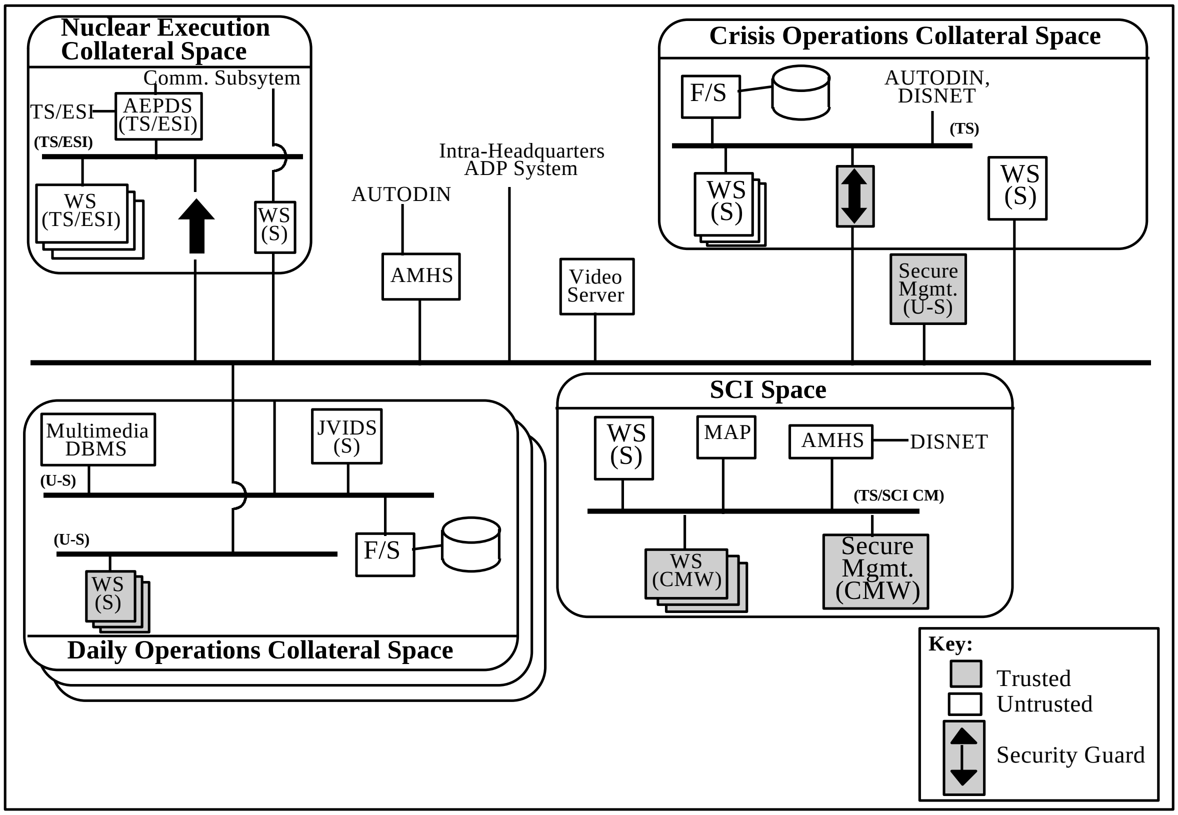


Figure 2-7.
Security Architecture – Concept Example

Standard Operating Procedure (SOP) Architecture . Virtually all workplaces are subject to policies and procedures that regulate activities and the use of resources. Ideally, these procedures should fit together to increase the efficiency of both systems and people. An SOP architecture documents the SOPs, policies, guidance, and instructions in force at an organization. This architecture assists the architect in determining what procedural requirements must be met, where the directions for fulfilling those requirements reside, and the precedence of one directive over another

An SOP architecture can be drawn from two different perspectives. First, SOP information can be overlaid on a simplified system architecture. For example, information on a classified processing SOP can be used to annotate classified systems and network segments. Components that will be replaced as part of an upgrade plan can be annotated with information on the origin and location of that plan. Restrictions on the number of workstations that can be attached to a particular network segment could be identified next to that segment in the drawing. The second type of SOP architecture focuses on the relationships among the policies themselves. This architecture could look very much like an organizational chart, with the high-level mission statement at the top flowing into programmatic guidance, technical plans, and detailed SOPs. See Figure 2-8.

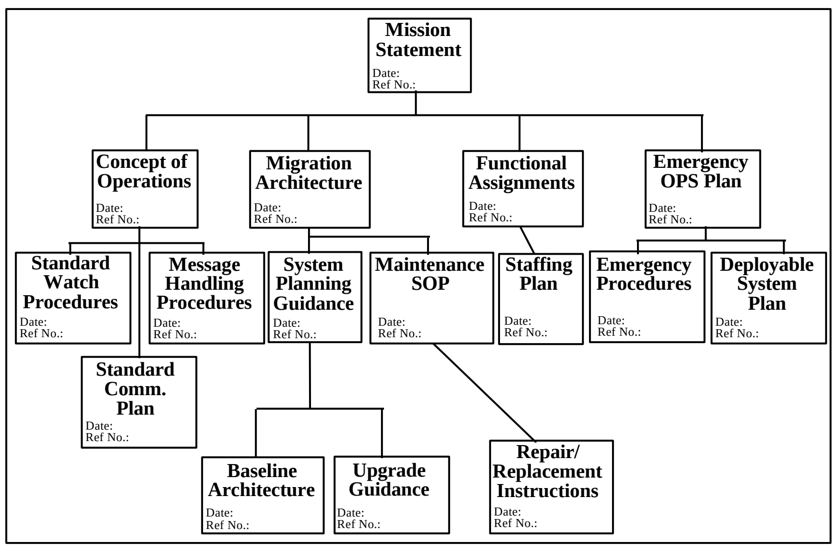


Figure 2-8.
Standard Operating Procedure Architecture – Concept Example

Template Architecture. A template architecture is one that has been designed and validated as appropriate for a specific purpose common to many organizations. This architecture supports the reuse of standard elements and configuration strategies, which expedites development at the user organization and encourages compliance with the standards shown in the template. A template architecture supplies the principal generic elements of a system, with the expectation that the local user can adapt the template and add details to tailor the architecture for a particular organization. For example, a Local Area Network (LAN) template may supply a basic configuration to be followed, with the number of clients changing from site to site as the template is modified to suit local requirements See Figure 2-9.

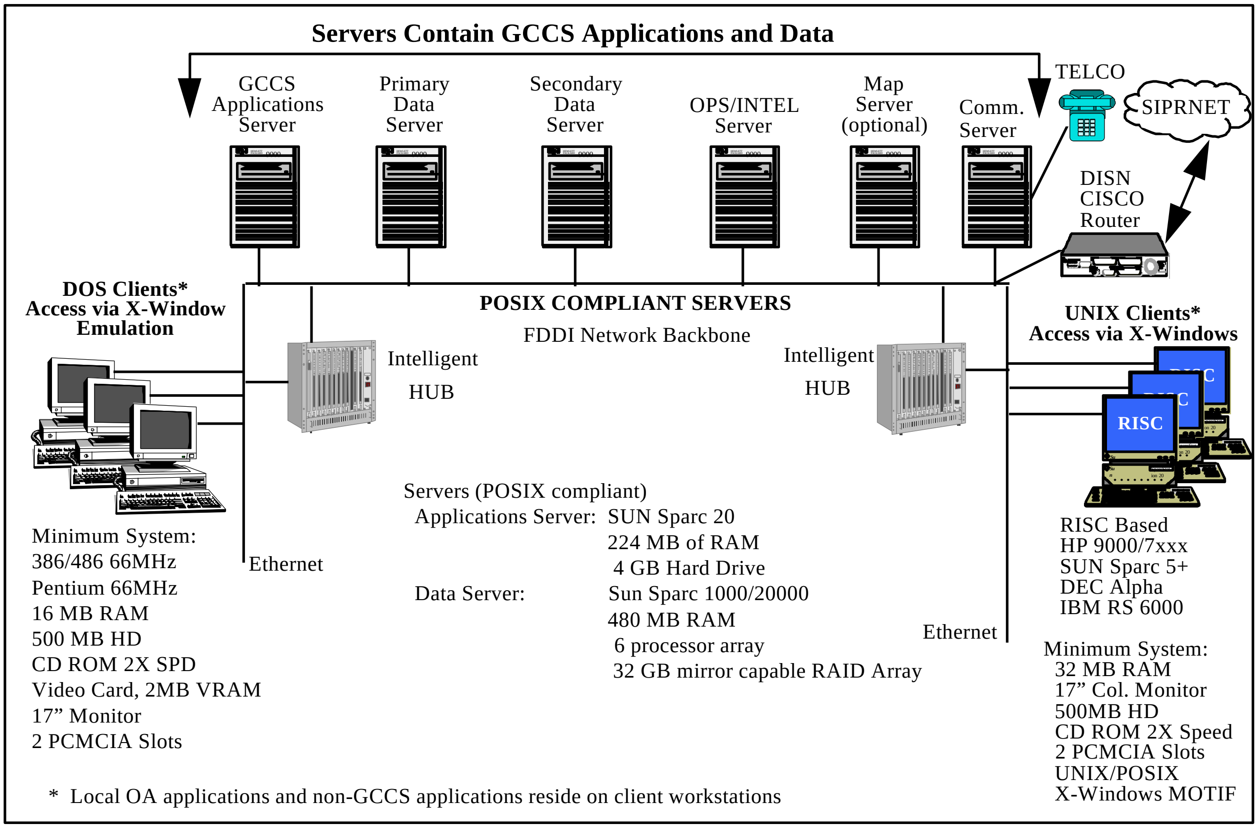


Figure 2-9.
Template Architecture – Concept Example

On-Line, Near-Realtime Architecture for Configuration Management . Military contingency operations frequently require rapid response and creative problem-solving – swapping workstations, changing networks, anything to get the job done. This architecture provides a way to track such changes on a near-realtime basis and maintain an accurate picture of system configuration.

Unlike the other architecture types discussed in this section, the on-line, near-realtime architecture for configuration management (CM) is dependent on the use of automated tools. A specialized software application continuously gathers data on systems and networks, while they are operating and automatically updates a graphic depiction of the architecture. In addition to simplifying CM, this architecture helps the user determine where recurring problems or deficiencies exist. The information gained is valuable input to acquisition planning and upgrade activities.

Network managers use the same types of network monitoring tools that architecture specialists use for the on-line CM architecture, but the way they use the tools and their programmatic aims are different. The architecture specialist is typically not involved in day-to-day network monitoring and troubleshooting operations. See Figure 2-10.

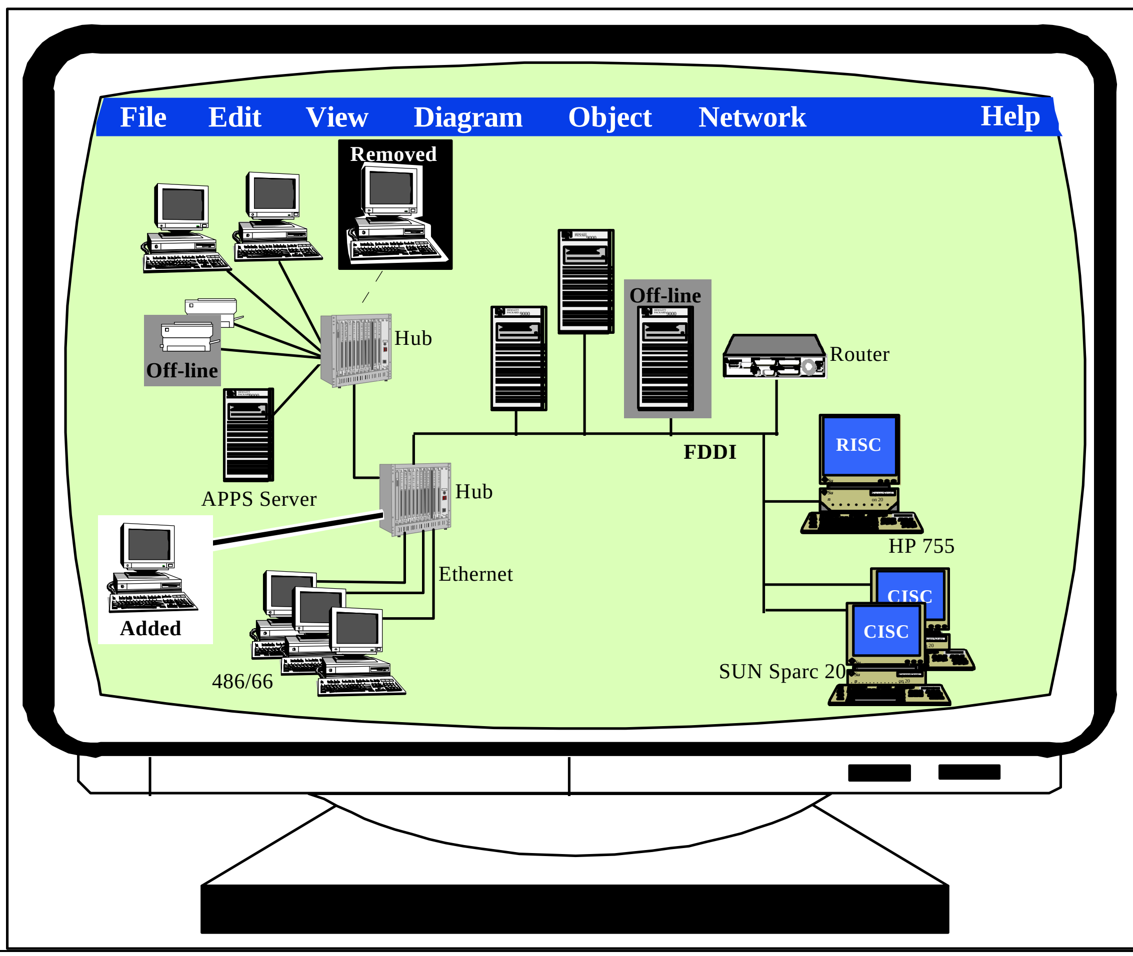


Figure 2-10.
On-Line, Near-Realtime Architecture for Configuration Management
– Concept Example

RELATED TERMS

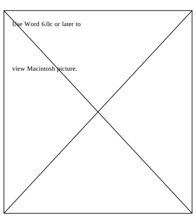
In addition to common terminology for the major types of architectures, standard terms have been defined for architecture drawings supporting GCCS implementation and for the GCCS elements depicted in those drawings.

Drawing Types

Drawing of Record. A drawing of record may be any type of architecture drawing and may show any level of detail desired. This drawing could be as detailed as the formal blueprint of a specific base, camp, or station, a record of a system or system, or simply a record of an agreement between two parties. Two characteristics make adrawing suitable to be a GCCS drawing of record:

- Preparation in the GCCS drawing format described in Section 3.
- The intent to use the drawing as a formal record of system configuration, planning data, or documentation of a concept.

The drawing officially becomes a drawing of record when it is assigned a record number by the GCCS Architect.



More information on drawings of record – how the submittal and approval process works and how it benefits you – appears in Section 4. Appendix H provides detailed instructions and an address for submitting drawings to the GCCS Architect.

Drawing of Acceptance. A drawing of acceptance is a drawing of record that has been coordinated and signed by the parties concerned. When a drawing of acceptance includes the signatures of senior approval authorities, it represents formal acceptance of an architecture at the Commander-in-Chief (CINC) level. Another reason for preparing a drawing of acceptance is to record an agreement between two organizations, each of which acknowledges acceptance by signature. For example, a drawing of acceptance may depict an agreed-upon transfer of equipment from Group A to Group B when replacements are received at some point in the future.

Drawings of acceptance and any supporting documentation are forwarded to the GCCS Architect for archiving.

Engineering Drawing. An engineering drawing is a detailed technical drawing or blueprint. It provides a level of technical detail appropriate for use by the architect or engineer to manage system design, installation, implementation, or maintenance. It is likely that you will generate a large number of engineering drawings to support day-to-day activity. These are not required to be sent to the GCCS Architect for archival.

Management Drawing. A management drawing is an architectural drawing tailored to support staff management, resource allocation, scheduling, budgeting, and other managerial activities.

Typically, an architecture drawing serves as a background, with managerial information contained in annotations or on overlays. For example, a management drawing might be based on a drawing of a network, with an overlay containing the names of personnel responsible for each segment along with the staff associated with that segment. A different view could highlight systems that require 24-hour staffing, or show those that have priority for power restoration and repairs.

As is the case with engineering drawings, management drawings do not need to be sent to the GCCS Architect unless the originator has a special need to preserve or formalize them.

Related Terms: GCCS Systems Components and Descriptors

Essential to the "GCCS vocabulary" that has been presented in this section are terms for the actual pieces of the system. The system components depicted in the GCCS architecture include the hardware, processors, operating systems, software applications, and network communications links used for local, as well as inter/intratheater connectivity. This section also includes definitions for some less tangible concepts that characterize system types and components.

Application Software. Software applications are programs that support specific computing needs. GCCS mission area application software (e.g., Joint Marines Command Information Strategy [JMCIS]/GCCS Status of Resources and Training System [GSORTS]) is tailored to the specific, unique needs of a particular end user or group of end users having access to the GCCS infrastructure. Support applications are those common programs that have more universal applicability across multiple user groups, such as word processing, briefing, or presentation applications. New GCCS mission area applications will be designed to access a common set of support applications and computing services for all users.

Hardware. The hardware consists of the primary physical machinery (personal computers [PC], workstations, servers, etc.), including the communications systems hardware, and the supporting peripheral equipment (e.g., hubs, routers, etc.) that collectively make up the GCCS infrastructure.

Infrastructure. Within the GCCS architecture, infrastructure includes processors, operating systems, software, and network communications links used for local, as well as inter/intratheater connectivity. Standards profiles are used to ensure interoperability among all of the network components. GCCS will ultimately be supported by the common Defense Information Infrastructure (DII) networks (circa 2000).

Legacy Systems. Legacy systems are current systems that provide key information management services that are candidates for integration, phase-out, upgrade, or replacement within the GCCS architecture. Validated legacy system workloads will be accommodated by the GCCS infrastructure as appropriate. Legacy systems will be analyzed to determine if they meet the GCCS requirements or if their functionality will be migrated to the future GCCS configuration.

Migration Systems. Migration systems are legacy systems that have been determined to accommodate a continuing and foreseeable future requirement and consequently have been identified for transitioning to GCCS. Migration systems may need to undergo transition to the standard technical environment of the GCCS, and therefore must migrate toward established standards for the GCCS infrastructure.

Open Systems Environment. The GCCS architecture is based on the principles of a comprehensive, enterprise-wide, open systems environment. This will ensure a system that implements open specifications for interfaces, services, and supporting formats to enable properly engineered applications software that ports readily, with minimal changes, across a wide range of systems, that interoperates with other applications on local and remote systems, and that interacts with users in a style that facilitates user portability.

Workstations. A workstation is a computer platform that provides user (i.e., client) interface to the GCCS infrastructure. The GCCS infrastructure is designed to accommodate both DOS-based (i.e., 386/486 PC machine) and UNIX-based (i.e., Reduced Instruction Set Chip [RISC]/Complex Instruction Set Chip [CISC] machine) client workstations.